


DEVICE AND METHOD FOR PROVIDING CUSTOMIZED ADVERTISEMENT ON WWW

Patent number: JP2000148675
Publication date: 2000-05-30
Inventor: MARK LANGHEINRICH; NAKAMURA ATSUYOSHI
Applicant: NIPPON ELECTRIC CO
Classification:
 - international: G06F15/00; G06F13/00; G06F17/30; G09G5/00
 - european: G06Q30/00A
Application number: JP19980317699 19981109
Priority number(s): JP19980317699 19981109

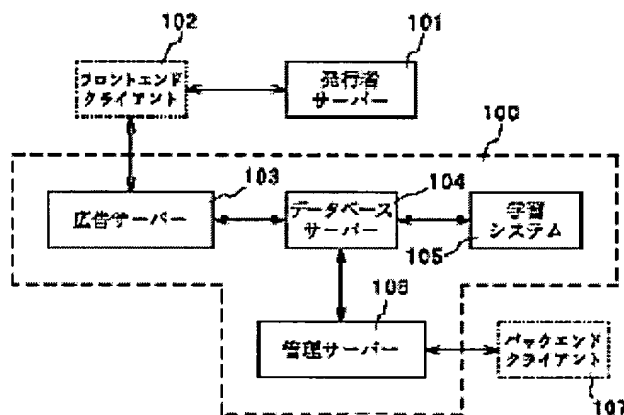
Also published as:

 US6654725 (B1)

Report a data error here

Abstract of JP2000148675

PROBLEM TO BE SOLVED: To provide a system and method for selecting and distributing customized advertisement on a world wide web (WWW) on the Internet. **SOLUTION:** An advertisement system 100 is provided with a data base server 104 for storing advertisement and campaign information and an advertisement server 103 for preparing electronic advertisement to be obtained by a client system. In this system 100, customizing process for customizing electronic advertisement to be distributed to each client system is executed. Then, a user is connected with a certain web site, and offered edited pages or a retrieved result list. In this system 100, customizing advertisement suited to page contents or retrieval topics is inserted into a page. Identifiable data are not collected during interaction with a user. An advertiser can designate display constraint on each advertisement. In this system 100, all unconstrained parameters are made suitable so that the click-through probability of the user is maximized.



Data supplied from the esp@cenet database - Worldwide

【外国語明細書】

1. Title of Invention:

A system and method for providing customized advertising on the World Wide Web

2. Claims:

1. An apparatus which provides electronic advertisements to a client system coupled to the apparatus, the apparatus comprising:

a database which stores advertisements and their campaign information;

an advertisement server which generates electronic advertisement available to the client system; and

means for performing a customization process which customizes the electronic advertisement to be delivered to each client system.

2. The apparatus according to claim 1 wherein the customization process determines the advertisement to be selected by using a set of customization parameters obtained from the client system.

3. The apparatus according to claim 1 wherein the database stores a plurality of advertisement titles, links and banner graphics, associated with a plurality of advertisements.

4. The apparatus according to claim 1 wherein the database furthermore stores a plurality of advertisement campaigns associated with a plurality of advertisements.

5. The apparatus according to claim 1 wherein the database stores a list of customization parameters for each campaign.

6. The apparatus according to claim 1 wherein the database stores display statistics for each advertisement and campaign each valid combination of customization parameters and their values.

7. The apparatus according to claim 1 where the database stores click-through statistics for each advertisement and each valid combination of customization parameters and their values.

8. A method of providing electronic advertisements to a client system, the method comprising steps of:

(a) decoding customization parameters embedded in a request from the client system;

(b) querying a database for a list of display probabilities for relevant values of the customization parameters;

(c) computing an overall display probability for the overall request; and

(d) selecting an advertisement according to the display probability.

9. The method according to claim 8 wherein the display probabilities queried in the querying step are pre-computed by a learning system.

10. The method according to claim 8 wherein values of the customization parameters decoded in the

decoding step are stored in the database.

11. The method according to claim 8 further comprising a step of

recording display and click-through for each set of customization parameters identified in the decoding step and selected advertisement in the selecting step in the database.

12. A method for querying a database for a list of display probabilities for relevant values of given customization parameters, the method comprising steps of:

providing a learning system which pre-computes the display probabilities; and

periodically updating display probabilities by the learning system.

13. The method according to claim 12, further comprising steps of:

(a) obtaining constraint data from the database;

(b) obtaining performance data from the database;

and

(c) obtaining customization parameter distributions from the database.

14. The method according to claim 12, further comprising a step of:

(a) storing a new list of display probabilities to the database.

15. The method of claim 12 wherein the list of display probabilities stored to the database is

calculated so as to maximize total click-through rate with respect to

(a) an estimated customization parameter distribution, and

(b) an estimated click-through probability for each pair of the customization parameter values and advertisements.

16. The method according to claim 15 wherein a total click-through rate is furthermore maximized under constraints of a daily impression rate for each advertisement.

17. The method according to claim 16 wherein the total click-through rate is furthermore maximized under custom display constraints set by an advertiser.

3. Detailed Description of Invention:

Technical Field to which the Invention Pertains:

The present invention relates to a system and method suitable for remote selection and delivery of customized contents such as advertisements, and particular to a system and method used to increase web advertisement response rates by providing customized advertisements shown as inline images and banners in web pages.

Prior Art:

Nowadays, the Internet is very popular with the consuming public and WWW (World Wide Web) pages on the Internet are considered powerful media for advertising. In its simplest form, web advertisement is directly

linked as fixed inline images into a web page. More flexible systems allow a separation of advertisement selection and placement, but offer only a random selection mechanism. This is possible since web documents do not contain images directly, but carry only a *reference* to the image itself. By having this reference not point to an actual image, but to an automated selection process, such a referenced inline image can be selected as late as the time the actual layout of the page is being made in the user's browser window. Moreover, each user who requests the page for display will potentially see a different inline image since the selection process is called for each user individually.

As an extension to this basic mechanism, many systems in use today (such as AdForce by AdForce, Inc. or AdKnowledge by AdKnowledge, Inc.) allow advertisers to specify targeting constraints that limit the display of an advertisement banner to certain conditions, such as the type of browser software used or the time of the day. Such a system first filters out all non-applicable advertisements given the condition of the current request for a banner. The remaining advertisements will then be selected randomly. Typical features include: type and version of browser software, operating system (OS), site originating the request, country, time of day, day of week. Several systems increasingly attempt to link such connection specific

information to user specific data such as age, gender, income, place of residence, etc.

Latest versions of advertisement selection systems (such as DART by DoubleClick, Inc.) offer simple click-boost mechanisms that, on top of the feature filtering mechanisms described above, keep statistics on how well each advertisement performed under the present conditions. Once a pool of available advertisements has been filtered out, the advertisement with the highest click-through will be selected. Others identify each user using a "cookie" (See Kristol, D. and Montulli, L. *RFC2109: HTTP state management mechanism*. Network Working Group, IETF, February 1997) (RFC = Request for Comments, and IETF = Internet Engineering Task Force) and limit the amount of times the same advertisement is shown to prevent "banner wearout". A cookie is a short piece of information, typically a user ID, that is sent by the server together with the requested page or image. The user's browser will store this information and resubmit it whenever the user requests a page or an image from the same server.

Finally, a few systems (such as the Accipiter AdManager from Engage or SelectCase for Ad Servers from Aptex, Inc.) combine neural network technologies (as described by Caid, W. et al. in United States Patent No. 5,619,709, *System and Method of context vector generation and retrieval*. April 8, 1997) with

individual user identification to create fully personalized advertisement placement, observing every single web page a user requests and thus accumulating an online interest dossier on each user.

Problem to be Solved by the Invention:

Many of the methods described above are too simple to take advantage of the just-in-time selection and delivery process of Web advertisement. Filtering techniques allow for a very precise targeting, but leave the task of selecting *whom* to target *what* advertisement to largely to the advertiser. This requires extended efforts on the advertiser side, who has to rely on countless statistics and demographic studies.

Although personalized advertisement delivery seems to solve this problem, the high amount of user monitoring clashes with an individuals need for privacy. Moreover, such intrusive techniques have not yet been proved to be effective in boosting click-through rates. However, one of the largest drawback of such systems using neural network technology is their inability to take display constraints, such as the minimum number of impressions to be shown for each advertisement, into account when selecting the best matching advertisement.

An objective of the present invention is improving the system for providing customized advertisement selection and delivery on the network.

Another objective of the present invention is improving the method for providing customized advertisement selection and delivery on the network.

Means for Solving Problem:

The first objective of the invention is achieved by an apparatus which provides electronic advertisement to a client system coupled to the apparatus, the apparatus comprising: a database which stores advertisements and their campaign information; an advertisement server which generates electronic advertisement available to the client system; and means for performing a customization process which customizes the electronic advertisements to be delivered to each client system.

The other objective of the present invention is achieved by a method of providing electronic advertisements to a client system, the method comprising the steps of: (a) decoding customization parameters embedded in a request from the client system; (b) querying a database for a list of display probabilities for relevant values of the customization parameters; (c) computing an overall display probability for the overall request; and (d) selecting an advertisement according to the display probability.

The other objective of the present invention is also achieved by a method for querying a database for a list of display probabilities for relevant values of given customization parameters, the method comprising

steps of: providing a learning system which pre-computes the display probabilities; and periodically updating display probabilities by the learning system.

The present invention is directed to a system serving customized advertisements on-demand over the Hypertext Transfer Protocol (HTTP). A user connects to the system indirectly by downloading a content page from a publisher which contains a reference to an inline image. This image is then requested by the browser, transparent to the user, directly from the system described here. The system uses a pre-computed, periodically updated table of display probabilities which prescribe a distribution for the available advertisements given the current conditions. By making a choice according to the obtained distribution the system will return a customized banner advertisement to the user's browser.

Mode for Carrying out the Invention:

As shown in FIG. 1, The advertising system 100 according to a preferred embodiment of the present invention consists of 4 main components, that is, an advertisement server (ad server) 103, a database server 104, a learning system 105 and a maintenance server (administration server) 106. All main components (servers 103, 104 and 106, and system 105) can be located on different machines in different physical locations, or even run on a single machine. The system interacts with a user's frontend client system 102 such

as but not limited to a system on which standard Web browser software runs. The frontend client system 102 originally interacts with a publisher's server 101. On the backend side advertiser's can use a backend client system 107 such as but not limited to a system on which a standard web browser software runs. The backend client system 107 is connected to the administration server 106, which offers access to the main database.

The advertisement server 103 handles incoming requests from clients to deliver customized advertisements. It contacts the database server 104 in order to obtain the data relevant to make an advertisement selection. It is to be noted that the database server 104 together with its database API (application program interface) 118 (FIG. 2) are not part of the present invention but can be any standard database system. The current embodiment uses a relational database product, but persons skillful in the art will realize that other forms of database systems can be used as well, provided they offer means to store the information necessary for the operation of the system. Once a selection is made, this selection will be logged with the database server. The learning system 105 periodically queries the logfiles from the database server 104 to obtain performance records of advertisements and adjusts a set of display weights accordingly. These weights are stored in the database server 104, where the advertisement server 103 can

access it during advertisement selection. Advertisers can contact a maintenance server 106 that allows direct access to the advertisement information stored in the database server 104.

FIG. 10 shows a sample interaction of a user with the publisher's site. After connecting to the publisher's site using a web browser 10 as shown in a window on a user's display screen, the user is presented with a search page 13. The page contains a dialog box 12 where the user can enter a keyword to search the publisher's database. Note that the initial search page 13 already features a banner advertisement 11 that has been selected and supplied by the advertisement server 103 (FIG. 1), not the publisher's server 101. After submitting the search word, FIG. 11 shows a result page 23 returned by the publisher's server 101 to the user's browser 10. The page contains another search box 22 as well as another advertisement 21 which was customized for the respective search word used.

FIG. 2 shows the main modules that comprise the advertisement server (ad server) 103. A Hypertext Transfer Protocol (HTTP) (see T. Berners-Lee, R. Fielding, and H. Frystyk, *RFC1945: Hypertext Transfer Protocol - HTTP/1.0* IETF, May 1996) decoding and encoding module 111 handles the communication from and to the user client. Typically, a standard web server such as NCSA's httpd (NCSA = National Center for

Supercomputing Application) or the Apache Web Server can be used. For persons skillful in the art it is easy to substitute one system for another, or even provide a custom implementation. Once a request has been decoded, one of three modules will handle the request: a module 112 for requests for image data, a module 113 for requests for HTML code, and a module 114 for click-through requests. These modules 112 to 114 will be explained in more detail below. Each module uses the advertisement server API (ad server API) 119 to perform the necessary computations and obtain relevant data from the database through the database API 118. The advertisement server API 119 offers functions in an advertisement data module 115 that provide access to advertisement data such as title, link URL (uniform resource locator) or image data; a selection module 116 that selects an advertisement ID (ad ID) to be displayed given a set of calling parameters; and a logging module 117 that provides logging and lookup functions for advertisement display and click-through statistics.

FIG. 3 shows a flow chart of the script that handles requests for an advertisement image. Upon invocation by the web server in step 1001, the script will first decode the parameters that have been passed to the script in step 1002. The Common Gateway Interface (CGI) (as defined by the NCSA) is a standard protocol that allows client and server applications to

exchange data over HTTP. CGI is implemented in almost all common web server implementations today, but persons skillful in the art will realize that it is easy to provide a custom implementation with similar support. The selection process can be shortcut with explicitly requesting a particular advertisement by its advertisement ID in step 1003. Otherwise the system tries to detect customization parameters in the request in step 1005. In the example shown in FIG. 11, the user's search word is a customization parameter, but it could also be a page ID or the name of the user's browser software. In case such information has been embedded into the request, the system will call the selection module 1006 to select a customized advertisement for the particular situation. If neither advertisement ID nor customization parameters are present, the system will simply obtain a list of currently active advertisements (i.e. advertisements that feature display constraints which do not prevent them from being shown under the current conditions) in step 1007 and select the advertisement with the highest required impression rate in step 1008. The impression rate of an advertisement is simply the number of times it should be shown on a certain web page or within a certain web site, and the amount of time left in the period it should be displayed in. This information is usually given by the advertiser and needs to be within the limits of total page accesses to the publisher's

web site the advertisement will be shown on. Once an advertisement id has been determined in step 1003 and the procedure of selection module 1006 has been performed, the system can then call the advertisement data module 115 (FIG. 2) for obtaining the actual image data in step 1009. After returning this information to the client at step 1010 (this of course involves adhering to proper CGI output specification) the system will log the impression of the particular advertisement in step 1011 and the customization parameter used (if any). In case no advertisement ID had been explicitly specified 1012, additional bookkeeping is necessary to synchronize click-throughs at a later time. Should some form of session ID be embedded in the request in step 1013 the system assumes that the corresponding hyperlink has the same session ID embedded (this would be done on the publisher's side) and simply logs the advertisement ID under the particular session ID in step 1014. Otherwise the system has to use other means of identifying a user, such as the IP (Internet Protocol) address of the connection, and log the display accordingly in step 1015. Persons skillful in the art will realize that other forms of identification can be used instead, such as an advertisement server assigned user ID transmitted via cookies. After storing log information, execution ends at step 1012.

FIGs. 9(a) and 9(b) illustrate the corresponding data flow between the user's web browser 102, the

publisher's content server 101 and the advertisement Server 103. FIG. 9(a) shows a case where the publisher has direct access to the advertisement server's API 119. After the user requests a page [1] from the Content Server 101, the server executes a script that will customize the requested page - for example by searching a database for a list of matching entries for a user query. In the process the publisher's script uses [2] the advertisement server selection API 119 to obtain an advertisement ID for the given customization parameters (for example the search keyword). The API 119 will query [3] to the database server and obtain the relevant information [4] from the database server 104, make a selection based on the customization parameter and return [5] the respective advertisement ID back to the calling script. This advertisement ID information can then be embedded into the HTML image tag that is placed on the results page where the advertisement will be displayed. After returning [6] the results back to the client, the user's web browser 102 will next try to download the referenced image data in the HTML page. Instead of pointing back to the Publisher's Content Server 101, the image tag will point to the advertisement server 103, which will be contacted [7] by the browser accordingly. As described in step 1003 in FIG. 3, the selection process 119 will find the embedded advertisement ID in the request [8] and directly contact [9] the database server 104 for

the raw image data (step 1009 in FIG. 3). The returned data [10] is handed back [11] to the advertisement server 103 and transmitted [12] to the user's web browser 102, where it can then be displayed to the user on the results page.

Alternatively, the system can perform advertisement selection and delivery in a single process, as shown in the data flow diagram in FIG. 9(b). After contacting [1], the content server 101 the corresponding script simple includes the customization parameters into the HTML image tag and returns [2] the results page to the user's web browser 102. The browser 102 again contacts [3] the advertisement server 103 for the actual image data, only this time including the customization parameters instead of the advertisement ID when starting the selection process [4]. As described in step 1005 in FIG. 3, the selection process 119 will find the embedded customization parameters to request and perform the selection after obtaining the relevant information [5], [6] from the database server 104. This advertisement ID information can then be used to directly contact [7] the database 104 for the raw image data (step 1009 in FIG. 3). Again, the returned data [8] is handed back [9] to the advertisement server 103 and transmitted [10] to the user's web browser 102, where it can then be displayed to the user on the results page. It is to be noted that in this case the chosen selection is at

the same time noted in the database 104 under a client-identifying attribute, such as the IP address, a user ID assigned via cookies, or a session ID that was embedded in the request by the originating publisher. This information will later be required for proper click-through handling (see below).

The advertisement server 103 also supports the usage of advanced HTML elements on the publisher side to facilitate state maintenance between advertisement display and click-through, alleviating the need for the above synchronization log file. FIG. 9(c) shows the dataflow for this case. After getting a request [1] from a client, the publisher's server 101 sends an HTML-embedding tag such as ILayer or IFRAME (see Raggett, D. and Le Hors, A. and Jacobs, I., *HTML 4.0 Specification*, The World Wide Web Consortium, 24 April 1998) together with the response [2], instead of the traditional image tag. The HTML-embedding tag points to the advertisement server 103 and contains the customization parameters that should be used by the advertisement system. Once the user's browser 102 encounters the embedding tags it will contact [3] the advertisement server 103 to fill in a snippet of HTML code at the position the original tag was placed in the output of the publisher's application. The advertisement server 103 will decode the request parameters as described in the flowchart in FIG. 4. Upon invocation at step 1101, the customization

parameters are decoded in step 1102, and whether the customization parameters are given or not is discriminated in step 1103. If found, the parameters are given to the selection module in step 1104. If no customization parameters are present, the system will simply obtain a list of currently active advertisements in step 1105 and select the advertisement with the highest required impression rate in step 1106. Once an advertisement ID has been determined, the system simply returns a predefined HTML block containing the selected advertisement ID as an embedded parameter to the HTML image tag in step 1107. These steps correspond to events [4] to [7] in FIG. 9(c). After returning [8], the HTML block back to the user's web browser 102, the advertisement server 103 will immediately be contacted [9] again by the browser 102 to fill in the actual image referenced in the HTML block. Events [10] through [14] are identical to events [8] through [12] of FIG. 9(a): given the actual advertisement ID the system simply follows the flowchart shown in FIG. 3 for delivering the actual advertisement graphic given the advertisement ID.

Once the user clicks on an advertisement, a surrounding hyperlink will first send the user's browser back to the advertisement server 103, where the click-through event is first logged and then a HTTP redirection command is issued that will point the user's browser to the web site of the corresponding

advertiser. The flowchart of this process is shown in FIG. 5. Upon invocation at step 1201, the script first decodes the CGI variables in step 1202. Both the process shown in FIGs. 9(a) and 9(c) will embed the advertisement ID directly in both the image and the surrounding hyperlink tag. This is first tested for in step 1203 and can eventually shortcut the process at step 1208. Otherwise, the system tries to find an embedded session ID in step 1205 that could optionally be embedded into both tags by the process described in FIG. 9(b). Should session information be available, the system looks up the corresponding advertisement ID in step 1206. Otherwise an alternative identification, such as the IP address is used in step 1207. In either case, the obtain advertisement ID is used to obtain the URL link information from the advertisement data Module in step 1208 and an HTTP redirection to the correct URL is sent back to the client in step 1209. Finally, the click-through event is logged together with the customization parameters used in step 1210 (this information can either be found in the request itself or in one of the logs).

The structure of the selection module 116 (called in step 1106, FIG. 4) is illustrated in FIG. 6. A decoding module 2001 extracts the customization parameters f_1, f_2, \dots, f_n . Using the extracted parameters, the database system 104 is queried through the database (DB) API 118 to provide a list of display

probabilities for all available advertisements in the system, given the values of each customization variable. The list of display probabilities is encoded and outputted by an encoding module 2003. For each value of a customization variable, this list of probabilities for each advertisement has to add up to 1.0, i.e. in every case the system has to be able to choose one and only one of the available advertisements. After compiling this display distribution for all advertisements in a relevancy computation module 2002, the system chooses a random advertisement according to the given probabilities (weight) 2005. Once an advertisement ID (ad ID) has been chosen, it is returned to the calling program (advertisement selection module 2004). Recalling the request description above, this can then either be used to directly query the database for image data (event [7] in FIG. 9(b)), or to embed an advertisement ID parameter into a request string for an image (event [6] in FIG. 9(a)) or inline HTML element (event [7] in FIG. 9(c)). It is up to the calling program to log the proper information in the database (FIG. 3, step 1011).

The learning process performed by the learning system 105 is completely separated from the on-line selection process. As shown in FIG. 7, a learning script 120 is called periodically which controls the input and output between the learning engine 121 and the rest of the system.

A flow chart of this process is shown in FIG. 8. The script will first contact the database server to obtain a list of current advertisements to show in step 1302. This list contains not only the ID of the advertisement, but also the requested daily impression rate and a list of manually configured customization parameters. The requested daily rate is computed as follows. Each advertisement is scheduled to be shown a certain number of times (called "impressions") over a certain period of time. This is called an advertisement "run". The daily rate is simply the number of impressions scheduled minus the number of impressions already shown, divided by the remaining time in the period this advertisement should be shown. The time can be expressed in days, hours or even seconds, as long as it is used consistently across all runs. The learning system will use only the relative sizes of the requested daily impression of all advertisements, not their absolute values. By using a finer shredded time parameter, the learning system is able to adjust the display probabilities more accurately in short time periods. The list of manually configurable customization parameters is simply an unordered list of attributes and their values together with a fractional value between and including 0.0 and 1.0. This fractional value can be set by the advertiser for each advertisement run to set minimum or maximum limits for the display probabilities the

learning engine will compute.

The second item obtained by the learning script in FIG. 8 in step 1303 is the current distribution of customization parameters. This list contains attributes and their values, each with a number indicating the frequency with which this attribute value has been encountered in the poll period. In it's simplest embodiment, the poll period is simply a time span such as 7 days or 48 hours indicating the amount of time the system should go back in time while analyzing the log files of the system to assemble this distribution list. Smaller time spans will enable the learning system to better adjust to momentarily changes in usage patterns (for example, when a current news topic significantly changes a query word distribution for a couple of days only), while greater time spans will smooth out such changes. As an alternative, this time span can be exchanged for a dampening factor that will be applied to events that lie further back in time. A higher dampening factor will decrease the impact of past events, while a lower factor will weigh past experience higher.

Finally, the learning script (FIG. 8) will assemble performance statistics in step 1304 using the same poll period and method used to obtain the customization parameter distribution in step 1303. This performance statistics will contain a data point for each advertisement and customization parameter

value combination. The data point consists of the number of impressions in the (possibly pro-rated) poll period and the number of click-throughs in the same period. Using this data, the learning system can evaluate the effectiveness of each value of the customization parameters in triggering a click-through for a certain advertisement. These three input files are collected by the learning script before calling the learning algorithm itself on these data files in step 1305. The present invention maps the problem of maximizing advertisement click-through in the presence of display constraints onto a constraint-problem that is solved using techniques from linear programming. Linear programming is a mathematical method for solving optimization tasks. It is frequently used to minimize costs or to maximize profits in manufacturing or transportation for example. To solve a problem, the real-world system is initially characterized by a limited number of system variables. With these system variables, a set of linear equations or inequalities are formulated to express restrictions on resources or requirements and to describe the quantity to be optimized. A solution to this model is any set of variables that does not violate the restricting system. An optimal solution is a solution with an optimal objective value. For purposes of describing and analyzing algorithms, the problem is often stated in the standard form

$$\min \{ c^T x, Ax = b, x \geq 0 \} \quad (1)$$

where $x \in R^n$ is the vector of unknowns, $c \in R^n$ is the cost vector, and $A \in R^{m \times n}$ is the constraint matrix. The feasible region described by the constraints is a polytope, or simplex, and at least one member of the solution set lies at a vertex of this polytope.

The present invention computes advertisement display probabilities for each customization parameter value so as to maximize the advertisement's click-through rate in the presence of display constraints. For simplification, we will discuss the learning system using a single customization parameter W only. For persons skilled in the art it is easy to extend this method to include additional customization parameters S, T, U, V , etc. Let A_1, \dots, A_m denote the list of advertisements and let W_1, \dots, W_n denote the list of customization parameter values. From the list of current advertisements to be shown, the learning engine obtains the requested daily impression rate h_j for each advertisement A_j . From the current distribution of customization parameters, the engine estimates the present probability k_i for each customization parameter value W_i . From the performance statistics file, the engine estimates the click-through rate $c_{i,j}$ for each customization parameter value W_i and each advertisement A_j . Then, the engine solves the problem of finding the display probabilities $d_{i,j}$ of advertisement A_j for

customization parameter value W_i so as to maximize total click-through rate

$$\sum_{i=1}^n \sum_{j=1}^m c_{i,j} k_i d_{i,j} \quad (2)$$

under the following constraints:

$$\sum_{i=1}^n k_i d_{i,j} = h_j, \quad (j=1, \dots, m) \quad (3)$$

$$\sum_{j=1}^m d_{i,j} = 1, \quad (i=1, \dots, n) \quad (4)$$

$$d_{i,j} \geq 0, \quad (i=1, \dots, n, j=1, \dots, m) \quad (5)$$

If there is a list of manually configurable customization parameters, the engine adds the constraints on $d_{i,j}$ to the problem. For example, if the advertiser wanted advertisement A_1 to be shown at least 30% of the time when customization parameter W had value W_0 and 10% of the time when the value is W_1 , the following two equations would be added by the system:

$$d_{0,1} \geq 0.3,$$

$$d_{1,1} \geq 0.1$$

Several analytical methods are known to find an optimal solution to a linear model (see Chvatal V., *Linear Programming* W.H. Freeman and Company, New York 1983, or Stephen J. Wright, *Primal-Dual Interior-Point Methods* SIAM 1996). The current embodiment uses a method known as the simplex algorithm, as described in James K. Strayer, *Linear Programming and Its Applications*, New York, 1989, Springer-Verlag, but persons skillful in the art will realize that from the

perspective of the overall system the actual algorithm is not important, as long as the input and output parameter are kept constant.

After having found an optimal solution for the problem, the learning system 105 returns a list of display probabilities $d_{i,j}$ for each advertisement A_j and customization parameter value W_i . This information is converted into an intermediate, database compatible format by the learning script in step 1306 (FIG. 8) and directly uploaded into the database where it can be instantaneously used by the selection process described in FIG. 6.

Effect of the Invention:

The overall system architecture allows distributed advertisement delivery. All components can either be at a single centralized location or on different machines and in different places. Although cookies can be used to help session identification, they are not necessary for operation. Most importantly, the system is able to automatically adapt to usage pattern. Advertisers simply have to register their advertisement with the system and can leave advertisement targeting to the automated learning system. However, the advertiser remains in full control by being able to specify an arbitrary number of display constraints. The system will attempt to maximize the click-through for each single advertisement by relying on past experience.

Performance can further be increased by grouping related advertisements into families and selecting among families instead of single advertisements.

4. Brief Description of Drawings:

FIG. 1 is a block diagram showing the main components of the advertising system according to a preferred embodiment of the present invention and their environment.

FIG. 2 is a block diagram of the advertisement server of the system shown in FIG. 1

FIG. 3 is a flow chart showing the procedure for raw image requests.

FIG. 4 is a flow chart showing the procedure for inline HTML requests.

FIG. 5 is a flow chart showing the procedure for click-through requests.

FIG. 6 is a block diagram of the selection module.

FIG. 7 is a block diagram of the learning system.

FIG. 8 is a flow chart showing the procedure of the learning script.

FIGs. 9(a), 9(b) and 9(c) are diagrams showing the different data flows between a publisher system, a user and the advertisement system.

FIG. 10 shows a sample interaction of a user with the publisher's content server containing a dynamically selected advertisement.

FIG. 11 shows a sample result from the

publisher's content server with a customized advertisement.

Description of the Reference Numerals:

100 Advertising system;

101 Publisher server;

102 Frontend client;

103 Advertisement server (Ad server);

104 Database server;

105 Learning system;

106 Administration server;

107 Backend client;

111 to 117 modules.

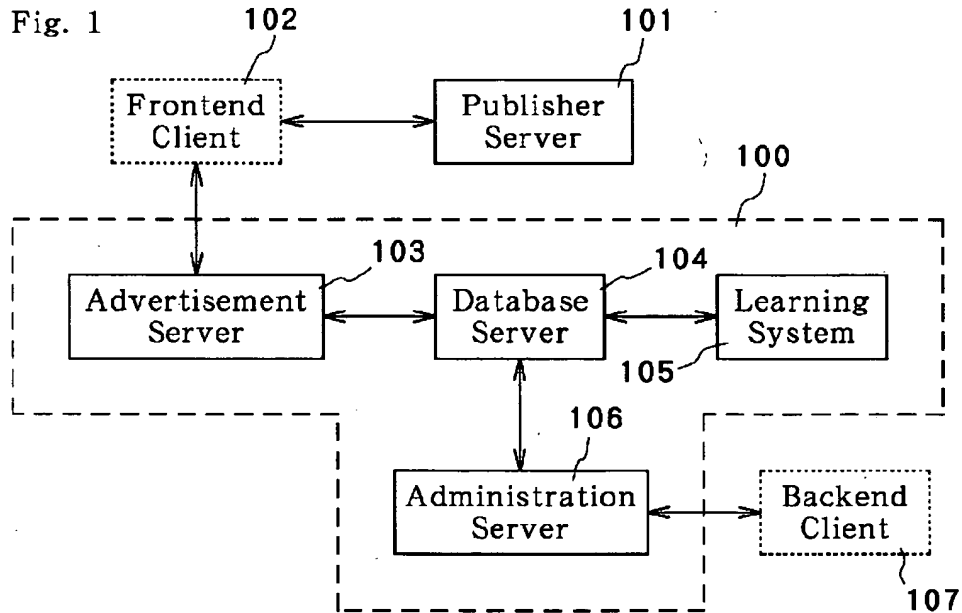


Fig. 2

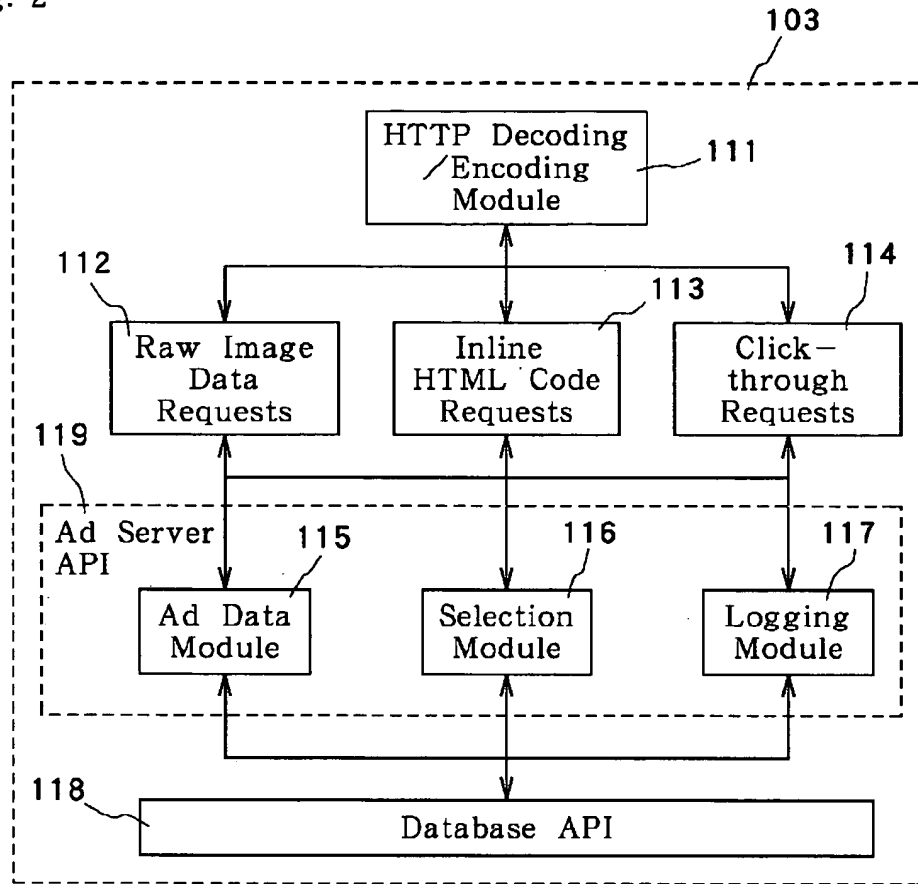


Fig. 3

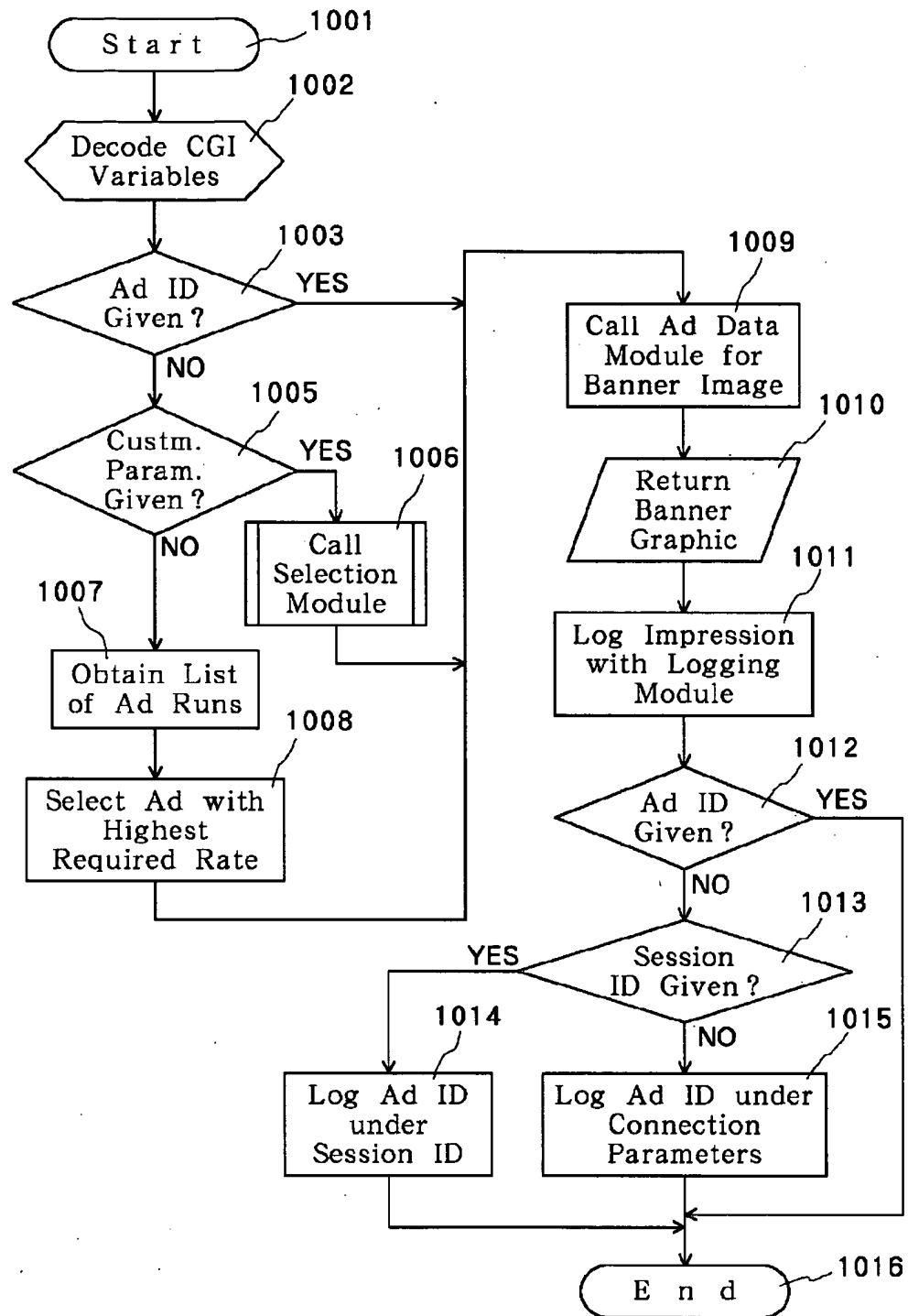


Fig. 4

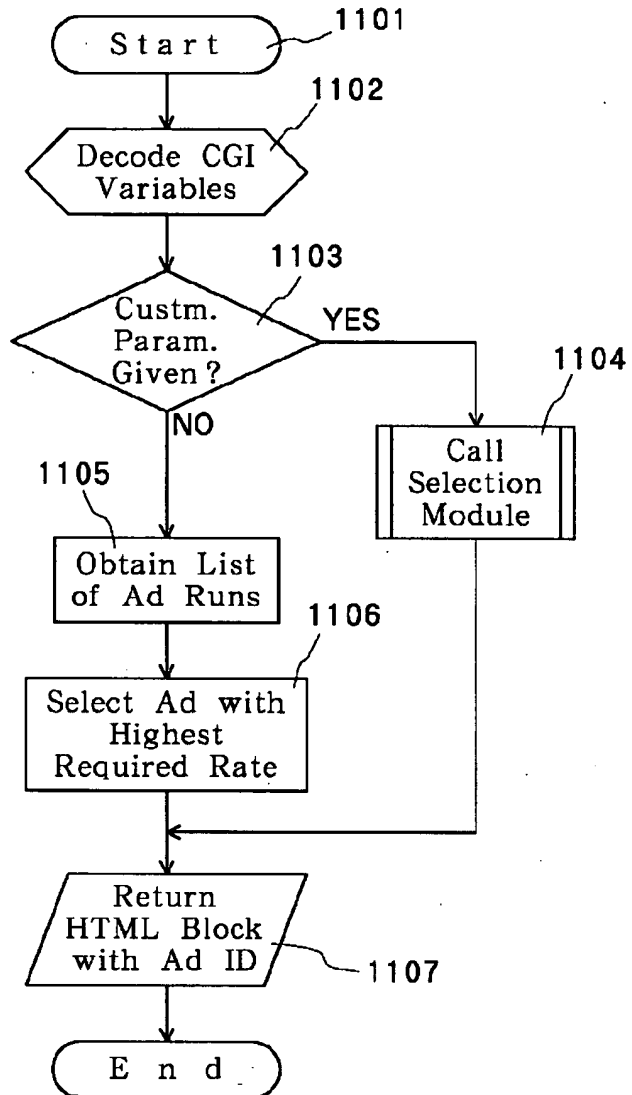


Fig. 5

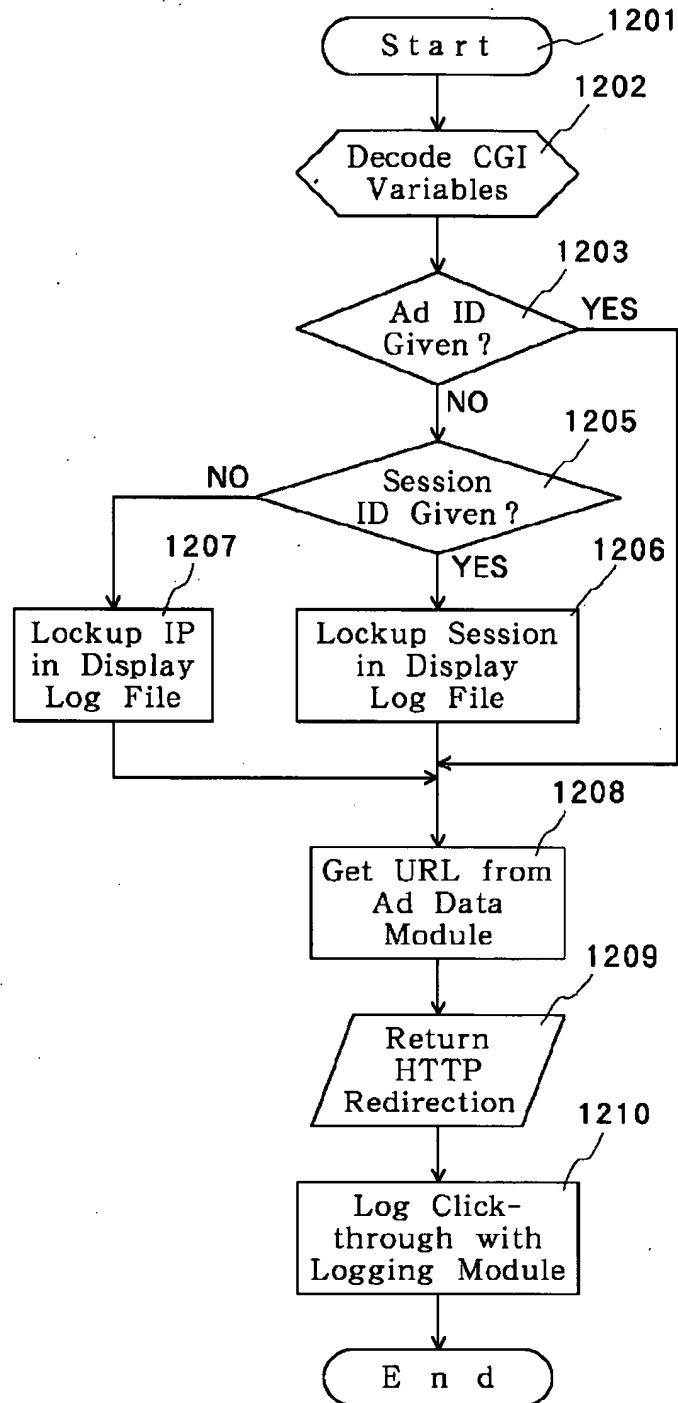


Fig. 7

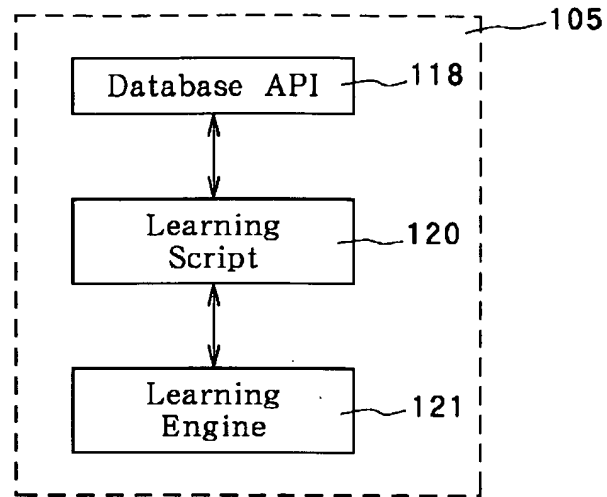


Fig. 8

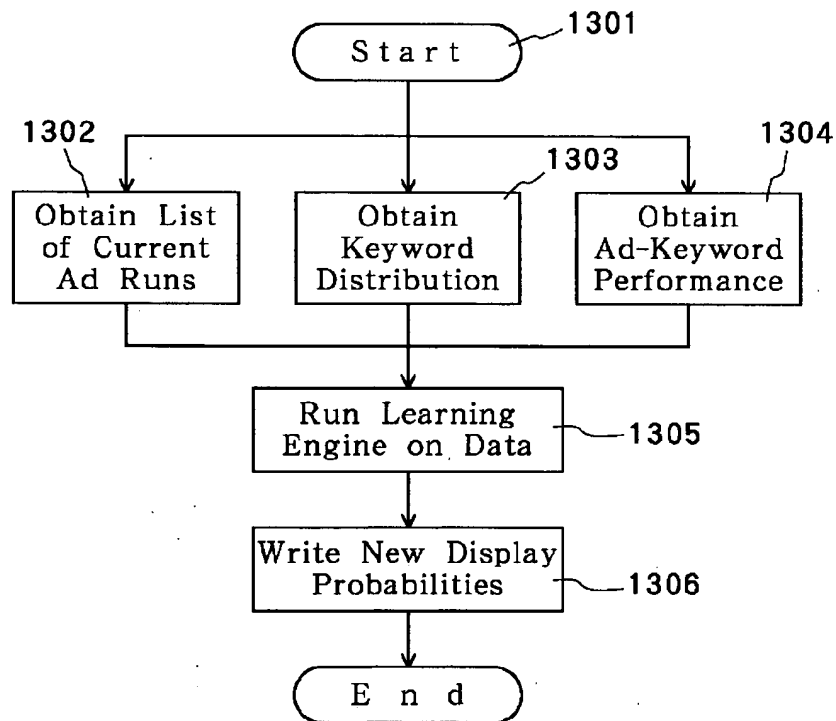


Fig. 9

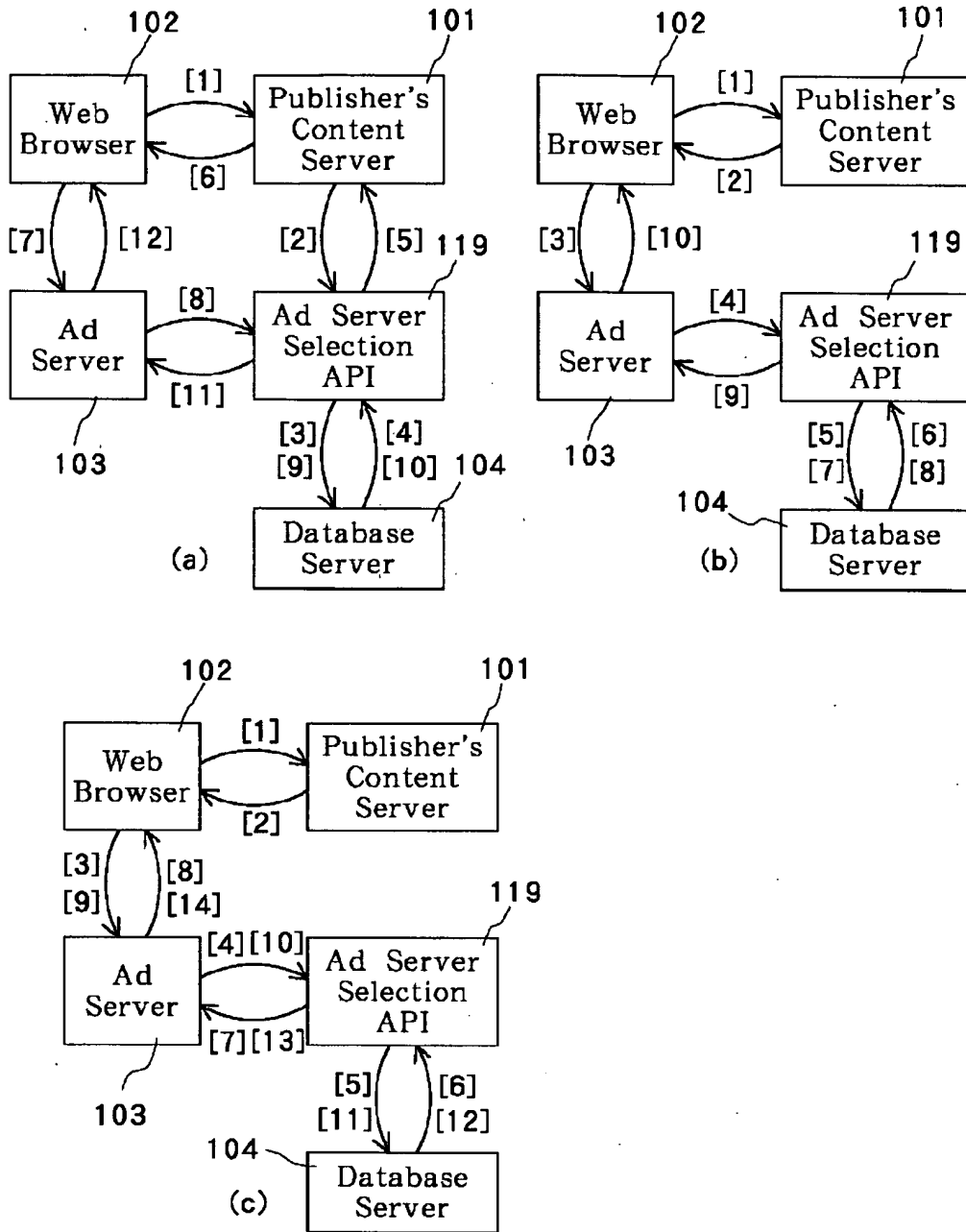


Fig. 10

ようこそAdvantage Ad Systemへ

このページは学習型インターネット広告システム (Advantage Ad) の動作を示すデモページです。

上記の広告は、Advantageによって選択されています。上記広告は、ランダムに選ばれていますが、もしもブラウザの「更新」ボタンを押してみてください。このシステムは以下の特徴を持ちます。

- 最も広告クリック率が高くなるように広告を予測選択
 - ユーザによる過去のクリック実績の考慮
 - 制約条件の満足
 - 表示回数保証
 - 特定キーワードの買い上げ
- ユーザのクリック結果を学習し、動的に広告表示確率を更新
- 使いやすい管理インターフェース

このページのデモでは、検索キーワード(Place)に対して7桁が入力した選択キーワードに最も適合する広告を自動的に選び出し、検索結果とともに表示します。以下の入力ボックスに検索キーワードを入れて、「Search」ボタンをクリック(あるいはリターンキーを押す)してください。

12. (検索キーワードを入力)

Search

Last modified: Wed Jul 29 14:04:25 JST 1998



1. Abstract:

Objective:

To provide a system and method for customized advertisement selection and delivery on the World Wide Web (WWW) upon the Internet.

Means for Solution:

The advertising system 100 has a database server 104 which stores advertisements and their campaign information, and an advertisement server 103 which generates electronic advertisements available to a client system. In the system, a customization process which customized the electronic advertisements to be delivered to each client system is performed. A user connects to a web site and is presented with an editorial page or a list of search results. The system inserts a customized advertisement into the page that matches the page content or search topic. No identifiable data is collected during the interaction with the user. Advertisers can specify display constraints for each advertisement. The system will adapt all unrestricted parameters in order to maximize the user's click-through probability.

2. Representative Drawing:

FIG. 1

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-148675

(P2000-148675A)

(43) 公開日 平成12年5月30日 (2000.5.30)

| (51) Int.Cl. ⁷ | 識別記号 | F I | テームコード* (参考) | |
|---------------------------|-------|---------------|--------------|-----------|
| G 0 6 F 15/00 | 3 1 0 | G 0 6 F 15/00 | 3 1 0 A | 5 B 0 7 5 |
| 13/00 | 3 5 4 | 13/00 | 3 5 4 D | 5 B 0 8 5 |
| 17/30 | | G 0 9 G 5/00 | 5 1 0 B | 5 B 0 8 9 |
| G 0 9 G 5/00 | 5 1 0 | G 0 6 F 15/40 | 3 1 0 F | 5 C 0 8 2 |
| | | | 3 7 0 Z | |

審査請求 有 請求項の数 1 O L 外国語出願 (全 51 頁)

(21) 出願番号 特願平10-317699

(22) 出願日 平成10年11月9日 (1998.11.9)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 マーク ラングハイブリッヒ

東京都港区芝五丁目7番1号 日本電気株式会社内

(72) 発明者 中村 篤祥

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100100893

弁理士 渡辺 勝 (外3名)

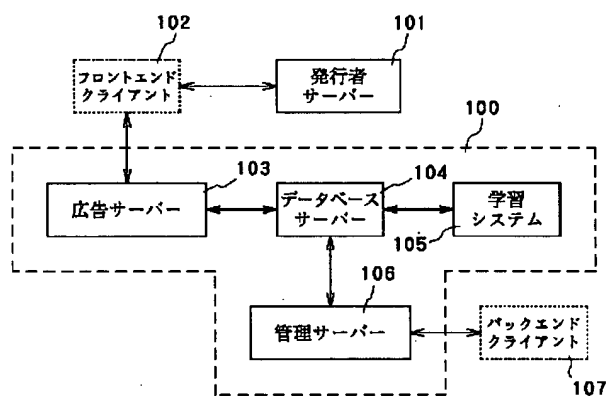
最終頁に続く

(54) 【発明の名称】 カスタマイズされた広告をWWW上で提供する装置及び方法

(57) 【要約】

【課題】インターネット上のワールド・ワイド・ウェブ (WWW) 上でのカスタマイズされた広告の選択及び配信のためのシステム及び方法を提供する。

【解決手段】 広告システム100は、広告とそのキャンペーン情報を格納するデータベースサーバー104と、クライアントシステムが入手できる電子広告を作成する広告サーバー103とを有する。このシステムでは、各クライアントシステムに配信されるべき電子広告をカスタマイズするカスタマイズプロセスが実行される。ユーザーは、あるウェブサイト に接続し、編集ページまたは検索結果リストを提供される。このシステムは、ページコンテンツまたは検索トピックに適合するカスタマイズ広告をページに挿入する。ユーザーとの対話中に、識別可能データは収集しない。広告者は、各広告について表示制約を指定できる。このシステムは、ユーザーのクリックスルー確率が最大となるように、全ての無制約パラメータを適合させる。



【特許請求の範囲】

【請求項1】 組み合わせられたクライアントシステムに電子広告を提供する装置であって、
広告及びそのキャンペーン情報を格納するデータベースと、
前記クライアントシステムが利用できる電子広告を作成する広告サーバーと、
各クライアントシステムに配信される前記電子広告をカスタマイズするカスタマイズプロセスを実行する手段と、を有する装置。

【請求項2】 前記カスタマイズプロセスが、クライアントシステムから得られるカスタマイズパラメータの組を用いることで、選択すべき広告を決定する請求項1に記載の装置。

【請求項3】 前記データベースが、複数の広告に関連する複数の広告タイトル、リンク及びバナー画像を格納する請求項1に記載の装置。

【請求項4】 前記データベースが、さらに、複数の広告に関連する複数の広告キャンペーンを格納する請求項1に記載の装置。

【請求項5】 前記データベースが、各キャンペーンについてのカスタマイズパラメータリストを格納する請求項1に記載の装置。

【請求項6】 前記データベースが、各広告及びキャンペーンと、カスタマイズパラメータ及びその値の各妥当な組み合わせとについての表示統計パラメータを格納する請求項1に記載の装置。

【請求項7】 前記データベースが、各広告と、カスタマイズパラメータ及びその値の各妥当な組み合わせとについてのクリックスルー統計データを格納する請求項1に記載の装置。

【請求項8】 クライアントシステムに電子広告を提供する方法であって、

(a) クライアントシステムからの要求に埋め込まれたカスタマイズパラメータを復号する段階と、

(b) カスタマイズパラメータの関連する値についての表示確率リストに関してデータベースに質問を行う段階と、

(c) 要求全体についての全体的な表示確率を計算する段階と、

(d) 前記表示確率に従って広告を選択する段階と、を有する方法。

【請求項9】 前記質問を行う段階で照会される表示確率が、学習システムによって予め計算されている請求項8に記載の方法。

【請求項10】 前記復号する段階で復号されるカスタマイズパラメータ値がデータベースに格納される請求項8に記載の方法。

【請求項11】 前記復号する段階で同定されたカスタマイズパラメータと前記選択する段階で選択された広告

との各組み合わせについての表示及びクリックスルーをデータベースに記録する段階をさらに有する請求項8に記載の方法。

【請求項12】 与えられたカスタマイズパラメータの関連する値についての表示確率のリストに関してデータベースに質問を行う方法において、

前記表示確率を予め計算する学習システムを準備する段階と、

前記学習システムにより定期的に表示確率を更新する段階と、を有する方法。

【請求項13】 (a) 前記データベースから制約データを取得する段階と、

(b) 前記データベースからパフォーマンスデータを取得する段階と、

(c) 前記データベースからカスタマイズパラメータ分布を取得する段階と、をさらに有する請求項12に記載の方法。

【請求項14】 (a) 前記データベースに表示確率の新たなリストを格納する段階をさらに有する請求項12に記載の方法。

【請求項15】 (a) 推定カスタマイズパラメータ分布と、

(b) カスタマイズパラメータ値及び広告の各ペアについての推定クリックスルー確率と、

に関して総クリックスルー率を最大とするように、前記データベースに格納された表示確率リストを計算する請求項12に記載の方法。

【請求項16】 各広告についての1日表示率の制約下で、総クリックスルー率がさらに最大化される請求項15に記載の方法。

【請求項17】 広告者が設定したカスタム表示制約下で、総クリックスルー率が最大化される請求項16に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、広告などのカスタマイズされたコンテンツの遠隔選択及び配信に好適なシステム及び方法に関し、特に、ウェブページでのインライン画像及びバナーとして示されるカスタマイズされた広告を提供することで、ウェブ広告の応答率を高めるのに使用されるシステム及び方法に関する。

【0002】

【従来の技術】今日、インターネットは大衆消費者に非常に人気があり、インターネット上のWWW（ワールド・ワイド・ウェブ）ページは広告のための強力な媒体であると考えられている。最も簡単な形でのウェブ広告は、ウェブページ内に固定されたインライン画像として直接リンクされている。より柔軟性の高いシステムは、広告の選択及び配置の分離ができるものであるが、ランダムな選択機構のみを提供するものである。これは、ウ

ェブドキュメントが画像を直接含むものではなく、画像自体に対する参照（レファレンス）のみを有することから可能である。この参照が実際の画像ではなく自動選択プロセスを指定するようにすることで、そのような参照インライン画像を、ユーザーのブラウザウィンドウでページの実際のレイアウトが形成されつつある時にまで遅らせて、選択することができる。さらに、その選択プロセスは各ユーザーについて個々に呼び出されるものであることから、そのページを表示するように要求を出している各ユーザーが、異なるインライン画像を見ることが可能である。

【0003】この基本的機構を拡大するものとして、今日使用されている多くのシステム（AdForce, Inc.（アドフォース社）によるAdForce、またはAdKnowledge, Inc.（アドノーレッジ社）によるAdKnowledgeなど）によって、広告者は、使用しているブラウザソフトの種類または1日の時刻など、広告バナーの表示を一定の条件に制限するターゲティング制約を指定することができる。そのようなシステムは最初に、バナーに関する現在の要求の条件を考慮して、全ての適用外広告を除去（フィルタリング）する。次に、残った広告を無作為に選択する。代表的な特徴としては、ブラウザソフトウェアの種類およびバージョン、オペレーティングシステム（OS）、要求を発するサイト、国、1日の時刻、曜日などがある。一部のシステムは、徐々に、そのような接続に固有な情報を年齢、性別、収入、居住地などのユーザー固有のデータにリンクさせる試みを行うようになっている。

【0004】最新バージョンの広告選択システム（DoubleClick, Inc.（ダブルクリック社）によるDARTなど）は、上記の特徴フィルター機構に加えて、現在の条件下で各広告がどの程度よく行われているかについての統計データを維持する簡単なクリックブースト機構を提供する。利用可能な広告の蓄積をフィルター処理したら、最も高いクリックスルー（click-through）を有する広告が選択される。他のものは、「クッキー(cookie)」(Kristol, D.とMontulli, L., RFC2109: HTTP state management mechanism, Network Working Group, IETF, February 1997参照) (RFC=Request for Comments, IETF=Internet Engineering Task Force) を使用して各ユーザーを確認し、同じ広告が表示される回数を制限して、「バナー飽き(wearout)」を防止するものである。クッキーとは、要求されたページまたは画像とともに、サーバーが送信する短い情報、代表的にはユーザーIDである。ユーザーのブラウザは、この情報を保存し、ユーザーが同一サーバーからのページまたは画像を要求する都度、それを再送信する。

【0005】最後に、いくつかのシステム（Engage（エンゲージ）からのAccipiter AdManager、またはAptex, Inc.（アプテック社）からのSelectCase for Ad Server

sなど）は、ニューラルネットワーク技術（ケイド(Caid, W.)らの米国特許5619709号「System and Method of context vector generation and retrieval」

（1997年4月8日）に記載のもの）と個々のユーザー識別とを組み合わせて、完全に個人化した広告配置を作り出して、いずれの単一ウェブページもユーザーの要求に従うようにし、そうすることで各ユーザーに関するオンライン関心事データ一式を蓄積する。

【0006】

【発明が解決しようとする課題】上記の方法の多くが単純すぎて、ウェブ広告のジャスト・イン・タイム選択及び配信を利用することができない。フィルタリング技術は、非常に正確なターゲティングを考慮したものであるが、広告者にとっては、誰に対してどの広告を中心的にターゲットとするかを選択する作業が残っている。それには、広告者側でかなりの労力が必要であり、広告者は無数の統計データおよび人口統計学的研究に頼らなければならない。

【0007】個人化された（パーソナライズされた）広告配信は、この問題を解決すると思われるが、ユーザーモニタリングの程度が大きいと、プライバシーを必要とする個人と対立することになる。さらに、そのような一方的技術が、クリックスルー率を高める上で有効であることは明らかになっていない。しかしながら、ニューラルネットワーク技術を用いるそのようなシステムの最大の欠点の一つは、最も適合する広告を選択する際に、各広告について表示すべき最小数の項目などの表示上の制約を考慮することができないという点である。

【0008】そこで本発明の目的は、ネットワーク上でのカスタマイズされた広告の選択及び配信を提供するシステムを改善することである。

【0009】本発明の別の目的は、ネットワーク上でのカスタマイズされた広告の選択及び配信を提供する方法を改善することである。

【0010】

【課題を解決するための手段】本発明の第1の目的は、組み合わされたクライアントシステムに電子広告を提供する装置であって、広告及びそのキャンペーン情報を格納するデータベースと、クライアントシステムが利用できる電子広告を作成する広告サーバーと、各クライアントシステムに配信される電子広告をカスタマイズするカスタマイズプロセス(customization process)を実行する手段と、を有する装置によって達成される。

【0011】本発明の他方の目的は、クライアントシステムに電子広告を提供する方法であって、(a) クライアントシステムからの要求に埋め込まれたカスタマイズパラメータ(customization parameter)を復号する段階と、(b) カスタマイズパラメータの関連する値についての表示確率リストに関してデータベースに質問を行う段階と、(c) 要求全体についての全体的な表示確率を計算

する段階と、(d) 表示確率に従って広告を選択する段階と、を有する方法によって達成される。

【0012】本発明の他方の目的は、与えられたカスタマイズパラメータの関連する値についての表示確率のリストに関してデータベースに質問を行う方法において、表示確率を予め計算する学習システムを準備する段階と、学習システムにより定期的に表示確率を更新する段階と、を有する方法によっても達成される。

【0013】本発明は、ハイパーテキスト・トランスファ・プロトコル (HTTP; Hypertext Transfer Protocol) によりオンデマンドでカスタマイズされた広告を提供するシステムに関する。ユーザーは、インライン画像に対する参照を含む発行者からのコンテンツページをダウンロードすることで、システムに間接的に接続する。次に、ユーザーにはトランスペアレントな(透過的な)ブラウザが、ここで記載するシステムからこの画像を直接要求する。このシステムは、予め計算されて定期的に更新される表示確率の表を使用する。この表示確率は、現在の条件を考慮して利用可能な広告についての分布を規定する。得られた分布に従って選択を行うことで、このシステムは、ユーザーのブラウザに対して、カスタマイズされたバナー広告を送り返す。

【0014】

【発明の実施の形態】図1に示したように、本発明の好ましい実施の形態による広告システム100は、4つの主要構成要素、すなわち、広告サーバー(アド・サーバー)103、データベースサーバー104、学習システム105及びメンテナンスサーバー(管理サーバー)106からなる。いずれの主要構成要素(サーバー103, 104, 106及びシステム105)も、異なった物理的位置にある異なった機械にあってもよく、あるいは単一の機械上で動作するものであってもよい。このシステムは、標準的なウェブブラウザソフトウェアが動作するシステムなど(それに限定されるものではない)のユーザーのフロントエンドクライアントシステム102と相互作用する(対話的に作用し合う)。フロントエンドクライアントシステム102は、もともと、発行者サーバー101と相互作用する。バックエンド側では、発行者(publisher)は、標準的なウェブブラウザソフトウェアが動作するシステムなど(それに限定されるものではない)のバックエンドクライアントシステム107を使うことができる。バックエンドクライアントシステム107は、管理サーバー106に接続されており、このサーバーは、メインデータベースへのアクセスを提供する。

【0015】広告サーバー103は、クライアントから入ってくる要求(リクエスト)を処理して、カスタマイズされた広告を配信する。このサーバーは、データベースサーバー104と交信して、広告選択を行う上で関連するデータを取得する。データベースサーバー104と

そのデータベースAPI(アプリケーションプログラムインターフェース)118(図2)が、本発明の一部ではなく、標準的なデータベースシステムであってもよいことは、留意すべき点である。本実施形態は、リレーショナルデータベース製品を使用しているが、当業者であれば、他の形態のデータベースも、それがシステムの動作に必要な情報を格納する手段を提供するのであれば、同様に使用可能であることが分かるであろう。選択がひとたび行われたなら、データベースサーバーにその選択が記録(ログ)されるであろう。学習システム105は、定期的にデータベースサーバー104からログファイルを照会して広告の実行記録(パフォーマンスレコード)を得て、それに従って、表示重みの組を調節する。それらの重みはデータベースサーバー104に格納され、そこで広告サーバー103が広告選択時にそれにアクセスすることができる。広告者(advertiser)は、データベースサーバー104に格納された広告情報に直接アクセスできるメンテナンスサーバー106と交信することができる。

【0016】図10は、ユーザーと発行者サイトとの対話(interaction)の例を示している。ユーザーのディスプレイ画面のウィンドウに示されるウェブブラウザ10を用いて発行者のサイトに接続すると、ユーザーには検索ページ13が示される。そのページは、発行者のデータベースを検索するためのキーワードをユーザが入力できるダイアログボックス12を含んでいる。最初の検索ページ13は、既に、発行者のサーバー101ではなく広告サーバー103(図1)によって選択・提供されたバナー広告11を表示している。検索語を送った後について、図11は、発行者のサーバー101からユーザーのブラウザ10に返送されてくる結果ページ23を示している。そのページは、別の検索ボックス22と、使用されたそれぞれ検索語についてカスタマイズされた別の広告21を含んでいる。

【0017】図2は、広告サーバー(アドサーバー)103を構成する主要モジュールを示している。ハイパーテキスト・トランスファ・プロトコル(HTTP)(T. Berners-Lee, R. Fielding及びH. Frystyk, RFC1945: Hypertext Transfer Protocol-HTTP/1.0 IETF, May 1996 参照)復号/符号化モジュール111は、ユーザークライアントから及びそのクライアントへの通信を扱う。代表的には、NC SAのhttpd(NC SA=National Center for Supercomputing Application)またはアパッチ・ウェブサーバー(Apache Web Server)などの標準的なウェブサーバーを使用することができる。当業者であれば、あるシステムを別のものに切り換えたり、あるいはカスタム仕様の実装を提供することは容易である。要求を復号したら、3つのモジュールのうちの一つがその要求を処理するだろう。すなわち、画像データについての要求に関してはモジュール112、HTMLコ

ードについての要求に関してはモジュール113、クリックスルー(click-through)要求についてはモジュール114である。これらモジュール112~114について、以下に詳細に説明する。各モジュールは広告サーバーAPI(アドサーバーAPI)119を用いて、必要な計算を行うとともにデータベースAPI118を介してデータベースから関連するデータを得る。広告サーバーAPI119は、タイトル、リンクURL(uniform resource locator)または画像データなどの広告データへのアクセスを提供する広告データモジュール115と、呼出しパラメータの組を考慮して表示すべき広告ID(アドID)を選択する選択モジュール116と、広告表示及びクリックスルーの統計データについてのログ機能と検索機能を提供するログモジュール117と、における機能を提供する。

【0018】図3は、広告画像についての要求を取り扱うスクリプトのフローチャートを示している。ステップ1001において、ウェブサーバーによる呼び出しがあると、スクリプトは、最初に、ステップ1002において、スクリプトに送られてきたパラメータを復号(デコード)する。コモンゲートウェイインターフェース(CGI; Common Gateway Interface)(NCSAによる定義による)は、クライアント及びサーバーのアプリケーションがHTTPを介してデータを交換できるようにする標準プロトコルである。CGIは、今日、ほとんど全ての一般的なウェブサーバー実装系で実装されているが、当業者であれば、同様のサポートを有するカスタム仕様の実装系を提供することが容易であることは理解できよう。この選択プロセスは、ステップ1003において広告IDによって特定の広告を明瞭に要求することで、ショートカットすることができる。そうしない場合、システムは、ステップ1005において、要求の中のカスタマイズパラメータの検出を試みる。図11に示した例では、ユーザーの検索用語がカスタマイズパラメータであるが、カスタマイズパラメータは、ページIDやユーザーのブラウザソフトウェアの名称であることもできると考えられる。そのような情報が要求に埋め込まれている場合、システムは、特定の状況向けのカスタマイズ広告を選択するよう、選択モジュール1006を呼び出す。広告IDもカスタマイズパラメータも存在しない場合、システムは、ステップ1007において、単に現在アクティブである広告(すなわち、現在の条件下で表示されるのを妨げない表示制約を示す広告)のリストを得て、ステップ1008において、その広告を最高要求表示率で選択する。広告の表示率(impression rate)とは、単純に、その広告があるウェブページ上またはあるウェブサイト内で表示されるべき回数と、その期間内でその広告が表示されるべき残り時間の量である。この情報は、通常、広告者によって提供され、広告が表示される発行者のウェブサイトへの全ページアクセスの範

囲内である必要がある。広告IDがステップ1003で決定され、選択モジュール1006の手順が行われたら、次にシステムは、ステップ1009において、実際の画像データを得るように、広告データモジュール115を呼び出す。ステップ1010においてその情報をクライアントに返した後(当然のことながらそれには、適切なCGI出力仕様の遵守が必要になる)、ステップ1011においてシステムは、特定の広告の表示及び使用したカスタマイズパラメータ(該当するものがある場合)の記録(ログ)を行う。広告IDが明瞭に特定されなかった場合(ステップ1012)、後で追加の記録(ブックキーピング)を行ってクリックスルーを同期させる必要がある。ステップ1013において要求に何らかの形のセッションIDが埋め込まれている場合、システムは、ステップ1014において、対応するハイパーリンクが同じセッションIDを埋め込まれていて(それは発行者側で行うことになると考えられる)、特定のセッションID下で単に広告IDの記録(ログ)を行うものと仮定する。そうでない場合、システムは、ステップ1015において、接続のIP(インターネットプロトコル; Internet Protocol)アドレスなどの他のユーザー識別手段を使用して、それに従って表示の記録(ログ)を行わなければならない。当業者であれば、クッキーを介して伝送されたユーザーIDによって指定された広告サーバーのような他の識別形態を代わりに使用することができることは、明らかであろう。ログ情報を保存した後、ステップ1012で実行処理が終了する。

【0019】図9(a)及び図9(b)は、ユーザーのウェブブラウザ102、発行者のコンテンツサーバー101及び広告サーバー103の間の対応するデータフローを図示している。図9(a)は、発行者が広告サーバーのAPI119に直接アクセスする場合を示している。ユーザーがコンテンツサーバー101からページを要求すると[1]、サーバーは、例えばユーザーの質問に適合する項目のリストについてデータベースを検索することで、要求されたページをカスタマイズするスクリプトを実行する。そのプロセスで発行者のスクリプトは、広告サーバー選択API119を用いて[2]、所定のカスタマイズパラメータ(例: 検索キーワード)に対する広告IDを得る。API119はデータベースサーバーに照会を行って[3]、データベースサーバー104から関連情報を得て[4]、カスタマイズパラメータに基づいて選択を行い、それぞれの広告IDを呼び出しスクリプトに返送する[5]。次に、その広告ID情報を、広告が表示される結果ページにあるHTML画像タグに埋め込むことができる。結果をクライアントに返送した後[6]、ユーザーのウェブブラウザ102は、次に、HTMLページで参照された画像データのダウンロードを試みる。従って画像タグは、発行者のコンテンツサーバー101を示すのではなく、ブラウザが交信する広告サ

ーバー103を示す[7]。図3のステップ1003に示したように、選択プロセス119は要求の中に埋め込まれた広告IDを見出し[8]、生画像データについてデータベースサーバー104と直接交信する[9]

(図3のステップ1009)。返送データ[10]は、広告サーバー103に送り返され[11]、ユーザーのウェブブラウザ102に伝送されて[12]、そこで次に、結果ページ上でユーザーに対して表示されることができる。

【0020】別法として、図9(b)のデータフロー図に示したように、システムは、単一のプロセスで広告の選択及び配信を行うことができる。コンテンツサーバー101と交信した後[1]、対応するスクリプトは、単に、HTML画像タグにカスタマイズパラメータを入れ、結果ページをユーザーのウェブブラウザ102に返送する[2]。ブラウザ102は、再度、実際の画像データについて広告サーバー103と交信する[3]。この時だけ、画像データには、選択プロセス開始時の広告IDではなく、カスタマイズパラメータが含まれている[4]。図3のステップ1005に示したように、選択プロセス119は、埋め込まれたカスタマイズパラメータを見出して、データベースサーバー104から関連情報を取得した後に、選択を要求・実行する([5]、[6])。次にこの広告ID情報を用いて、生画像データについてデータベース104と直接交信することができる[7](図3のステップ1009)。やはり、返送データ[8]は広告サーバー103に送り返され[9]、ユーザーのウェブブラウザ102に伝送され[10]、そこでそのデータを結果ページで上ユーザーに表示することができる。その場合、行った選択が、同時に、IPアドレス、クッキーを介して割り当てられたユーザーIDまたは元の発行者によって要求中に埋め込まれたセッションIDなどのクライアント識別属性の下で、データベース104中に書き込まれていることは、留意すべき点である。その情報は、後に、適切なクリックスルーの取り扱い(下記参照)に必要となる。

【0021】広告サーバー103は、さらに、発行者側での最新のHTML要素の利用をサポートして、広告表示とクリックスルーの間の状態メンテナンスを促進し、上記の同期化ログファイルの必要性を小さくする。図9(c)は、その場合のデータフローを示している。クライアントからの要求を受けて[1]、発行者のサーバー101は、従来の画像タグではなく、応答とともにI L A Y E RまたはI F R A M E (Raggett, D., Le Hors, A. 及びJabocs, I., HTML 4.0 Specification, The World Wide Web Consortium, 24 April 1998参照)などのHTML組み込みタグを送信する[2]。HTML組み込みタグは、広告サーバー103を示すものであり、広告システムが使用すべきカスタマイズパラメータを含む。ユーザーのブラウザ102が組み込みタグに接すると、ブ

ラウザは、広告サーバー103と交信して[3]、発行者のアプリケーションの出力において元のタグがあった箇所で、HTMLコード断片を書き入れる。広告サーバー103は、図4のフローチャートに示したように、要求パラメータを復号する。ステップ1101での呼び出しに応じて、ステップ1102においてカスタマイズパラメータが復号(デコード)され、ステップ1103においてカスタマイズパラメータが与えられているか否かが識別される。パラメータが見出されたら、ステップ1104において、そのパラメータが選択モジュールに出力される。カスタマイズパラメータがない場合、システムは、単に、ステップ1105において現在アクティブな広告のリストを得て、ステップ1106において、最も高い要求表示率の広告を選択する。広告IDが決定されたら、システムは、単に、ステップ1107において、HTML画像タグに対する組み込みパラメータとして特定の広告IDを含む所定のHTMLブロックを返送する。これらのステップは、図9(c)のイベント[4]～[7]に相当する。HTMLブロックをユーザーのウェブブラウザ102に返送した後[8]、広告サーバー103は、直ちに、ブラウザ102によって再交信して[9]、HTMLブロックで参照された実際の画像を書き込む。イベント[10]～[14]は図9(a)のイベント[8]～[12]と同一である。すなわち、実際の広告IDを考慮して、システムは、単に、図3のフローチャートに従って、広告IDを考慮した実際の広告画像を配信する。

【0022】ユーザーが広告上でクリックすると、周囲のハイパーリンクが最初にユーザーのブラウザを広告サーバー103に送り戻し、そこでクリックスルー事象が最初に記録(ログ)され、次にHTTPリダイレクションコマンドが出される。そのコマンドは、ユーザーのブラウザに、対応する広告者のウェブサイトを示すものである。このプロセスのフローチャートを図5に示してある。ステップ1201において呼び出しがあると、スクリプトは、最初に、ステップ1202においてCGI変数を復号(デコード)する。図9(a)及び図9(c)に示した両方のプロセスが、画像及び周囲のハイパーリンクタグの両方に広告IDを直接組み入れる。ステップ1203では最初に広告IDが調べられ、広告IDは結果としてステップ1208までプロセスをショートカットすることができる。それ以外の場合、システムは、ステップ1205において、図9(b)に示したプロセスによって両方のタグに組み込まれているかも知れない組み込みセッションIDを見出そうと試みる。セッション情報が入手できる場合、システムは、ステップ1206において対応する広告を調べる。入手できないときは、ステップ1207において、IPアドレスなどの別の識別手段が用いられる。いずれの場合も、得られた広告IDを用いて、ステップ1208において広告データモジュールか

らURLリンク情報を得て、ステップ1209において正しいURLへのHTTPリダイレクションがクライアントに送り返される。最終的に、ステップ1210において、使用したカスタマイズパラメータとともに、クリックスルー事象が記録（ログ）される（この情報は、要求自体またはいずれかのログで見出すことができる）。

【0023】選択モジュール116の構造（図4のステップ1106での名称）を図6に示してある。復号モジュール2001は、カスタマイズパラメータ f_1, f_2, \dots, f_n を抽出する。抽出したパラメータを用いて、データベース（DB）API118からデータベースシステム104に照会が行われて、各カスタマイズ変数の値を考慮して、システムにおける全ての利用可能広告についての表示確率リストが得られる。表示確率リストは、符号化モジュール2003によって符号化され出力される。カスタマイズ変数の各値について、各宣伝に関するこの確率リストは合計で1.0とならなければならない。すなわち、いずれの場合も、システムは、1個のみの利用可能な広告を選択することができなければならない。関連性計算モジュール2002で全ての広告についてのこの表示分布を編集した後、システムは、与えられた確率（重み）2005に従って無作為に広告を選択する。広告ID（アドID）が選択されると、それが呼び出しプログラムに送り返される（広告選択モジュール2004）。上記の要求についての説明に戻ると、次に広告IDを用いて、画像データについてデータベースに直接照会を行うか（図9(b)のイベント[7]）あるいは画像についての要求文字列（図9(a)のイベント

[6]）またはインラインHTML要素（図9(c)のイベント[7]）に広告IDパラメータを組み入れることができる。データベースへの適切な情報の記録は、呼び出しプログラムが担当する（図3、ステップ1011）。

【0024】学習システム105によって行われる学習プロセスは、オンライン選択プロセスから完全に分離している。図7に示したように、学習スクリプト120が定期的に呼び出されて、学習エンジン121とシステムの残りの部分との間の入出力を制御する。

【0025】このプロセスのフローチャートを図8に示してある。最初に、ステップ1302において、スクリプトは、データベースサーバーと交信して、示すべき現在の広告のリストを得る。このリストには、広告のIDだけでなく、要求される1日表示率およびマニュアル操作で設定されたカスタマイズパラメータのリストも含んでいる。要求される1日表示率は、次のように計算される。各広告は、ある一定期間で一定の回数（「表示（impression）」と称する）、表示されるよう予定されている。それを広告「実行（run）」と称する。1日表示率は、単に、予定表示数から既に示した表示数を引き、それをその広告が出されるべき期間の残り時間で割ったも

のである。その時間は、全ての実行で一貫して使用されるのであれば、日数、時間数または秒数として表すことができる。学習システムは、全広告の要求1日表示の相対的大きさのみを使用するものであって、それらの絶対値は使用しない。より細分化された時間パラメータを用いることで、学習システムは短期間でより正確に表示確率を調節することができる。マニュアル操作で設定可能なカスタマイズパラメータのリストは、単に、属性と0.0以上1.0以下の小数値でのその値の無秩序なリストである。その小数値は、学習エンジンが計算する表示確率について下限または上限を設定するために、各広告実行について広告者が設定することができる。

【0026】ステップ1303において図8の学習スクリプトによって得られる第2の項目は、カスタマイズパラメータの現在の分布である。このリストは、属性とその値を含み、それぞれに、その属性値がポーリング期間に出てきた頻度を示す数字を有している。最も簡単な実施形態では、そのポーリング期間は、単に、システムのログファイルを解析してその分布リストを組み立てながらシステムがちょうどよい時に戻るべき時間量を示す、7日間または48時間などの時間スパンである。時間スパンが短いほど、学習システムは使用パターンにおける時間的変更をより良好に調節することができるが（例えば、現在のニューストピックがわずか2日間の間に質問語分布を大幅に変える場合）、時間スパンが長くなるとそのような変化が滑らかになる。別法として、この時間間隔を、時間をさらにさかのぼった事象に適用される減衰係数(dampening factor)に交換することができる。減衰係数が高いほど、過去の事象の影響が低下し、係数が小さいほど過去の経験の比重が大きくなる。

【0027】最後に学習スクリプト（図8）は、ステップ1304において、ステップ1303においてカスタマイズパラメータ分布を得るのに用いたのと同じポーリング期間及び方法を用いて、パフォーマンス統計データを集める。このパフォーマンス統計データは、各広告及びカスタマイズパラメータ値の組み合わせについてのデータ点を含む。そのデータ点は、（場合によっては比例配分）ポーリング期間における表示数及び同期間におけるクリックスルー数からなる。このデータを用いて学習システムは、ある広告についてのクリックスルーを起こさせる上でのカスタマイズパラメータの各値の有効性を評価することができる。学習スクリプトはこの3つの入力ファイルを回収してから、ステップ1305において、それらデータファイルでその学習アルゴリズム自体を呼び出す。本発明は、表示制約の存在下での広告クリックスルー最大化の問題を、線形プログラミングからの方法を用いて解決される制約問題に変える。線形プログラミングは、最適化作業の解を得るための数学的方法である。それは非常に多くの場合、例えば製造もしくは輸送におけるコスト低減あるいは収益最大化のために使用

される。問題の解を得るために、まず、実世界システムが限られた数のシステム変数によって特徴づけられる。それらのシステム変数を用いて、1組の一次方程式または不等式が定式化されて、リソースまたは必要条件に関する制限を表わし、最適化すべき量を記述する。このモデルに対する解は、制限システムに違反しない変数の任

$$\min \{c^T x, Ax = b, x \geq 0\}$$

式中、 $x \in R^n$ は未知数のベクトルであり、 $c \in R^n$ はコストベクトルであり、

【0029】

【外1】

$$A \in R^{m \times n}$$

は制限行列である。上記条件式によって示されている実行可能領域は、ポリトープ（多面体）またはシンプレックス（単体）であり、解集合の1以上の要素がこのポリトープの頂点にある。

【0030】本発明は、各カスタマイズパラメータ値について広告表示確率を計算して、表示制約の存在下に、広告のクリックスルー率を最大とするものである。簡略化のため、1個のカスタマイズパラメータWのみを用いる学習システムについて考察する。当業者であれば、その方法を拡大して、別のカスタマイズパラメータS, T, U, Vなどを含めることは容易である。A₁, ..., A_nを広告リストとし、W₁, ..., W_nをカスタマイズパラメータ値とする。示されるべき現在の広告リストから、学習エンジンは、各広告A_jについての要求1日表示率h_jを得る。現在のカスタマイズパラメータ分布から、エンジンは各カスタマイズパラメータ値W_iについての確率k_iを計算する。パフォーマンス統計データファイルから、エンジンは各パラメータ値W_iおよび各広告A_jについてのクリックスルー率c_{i,j}を計算する。次にエンジンは、カスタマイズパラメータ値W_iについての広告A_jの表示確率d_{i,j}を見出すという問題を解いて、総クリックスルー率

【0031】

【数2】

$$\sum_{i=1}^n \sum_{j=1}^m c_{i,j} k_i d_{i,j} \quad (2)$$

を、下記の制約下に最大とする。

【0032】

【数3】

$$\sum_{i=1}^n k_i d_{i,j} = h_j, \quad (j=1, \dots, m) \quad (3)$$

$$\sum_{j=1}^m d_{i,j} = 1, \quad (i=1, \dots, n) \quad (4)$$

$$d_{i,j} \geq 0, \quad (i=1, \dots, n, j=1, \dots, m) \quad (5)$$

マニュアル操作で構成可能なカスタマイズパラメータの

意の組である。最適解は、最適対象値を有する解である。アルゴリズムについて説明し、それを分析するため、その問題は下記の標準的な形で記述される場合が多い。

【0028】

【数1】

(1)

リストがある場合、エンジンはその問題にd_{i,j}に関する制約を加える。例えば、カスタマイズパラメータWが値W₀を有する場合は時間の30%以上、値がW₁である場合は時間の10%で広告者が広告A₁を表示したい場合、下記の2つの式をシステムに加えることになると考えられる。

【0033】

【数4】

$$d_{0,1} \geq 0.3,$$

$$d_{1,1} \geq 0.1$$

いくつかの解析方法によって、線形モデルに対する最適解が見出されることがわかっている（Chvatal V., Linear Programming, W.H. Freeman and Company, New York 1983、または、Stephen J. Wright, Primal-Dual Interior-Point Methods, SIAM 1996を参照）。本実施形態は、シンプレックスアルゴリズムとして知られる方法（James K. Strayer, Linear Programming and Its Applications, New York, 1989, Springer-Verlagに記載の方法）を用いているが、当業者であれば、入力および出力パラメータが一定に保たれている限り、システム全体の観点からすると、実際のアルゴリズムは重要ではないことは理解できよう。

【0034】問題に対する最適解を求めた後、学習システム105は各広告A_j及びカスタマイズパラメータ値W_iについての表示確率d_{i,j}のリストを返送する。ステップ1306（図8）において、その情報は、学習スクリプトによって、中間形式のデータベース互換書式に変換され、データベース中に直接アップロードされて、そこで直ちに、図6に示した選択プロセスによって使用することができる。

【0035】

【発明の効果】システム全体のアーキテクチャによって、分散構成での広告配信が可能となる。全ての構成要素が、単一の集中化された箇所にあってもよく、あるいは異なる機械と異なる場所にあってもよい。クッキーを用いてセッション識別に役立てることができるが、それは動作には必要ない。最も重要な点として、このシステムは、自動的に使用パターンを適合させることができる。広告者は、単に、システムに広告を登録するだけでよく、広告ターゲティングを自動学習システムに任せたままることができる。しかしながら広告者は、任意

の数の表示制約を特定できることによって、完全な制御を行う地位にとどまっている。システムは、過去の経験に基づいて、各1個の広告についてのクリックスルーを最大にしようと試みる。関連する広告を族に分類分けして、個々の広告ではなく族間での選択を行うことで、パフォーマンスをさらに増加させることもできる。

【図面の簡単な説明】

【図1】本発明の好ましい実施形態による広告システムの主要構成要素及びその環境を示すブロック図である。

【図2】図1に示したシステムの広告サーバーのブロック図である。

【図3】生画像要求に対する処理手順を示すフローチャートである。

【図4】インラインHTML要求に対する処理手順を示すフローチャートである。

【図5】クリックスルー要求に対する処理手順を示すフローチャートである。

【図6】選択モジュールのブロック図である。

【図7】学習システムのブロック図である。

【図8】学習スクリプトの処理手順を示すフローチャートである。

【図9】(a), (b)及び(c)は、発行者システム、ユーザー及び広告システムとの異なったデータフローを示す図である。

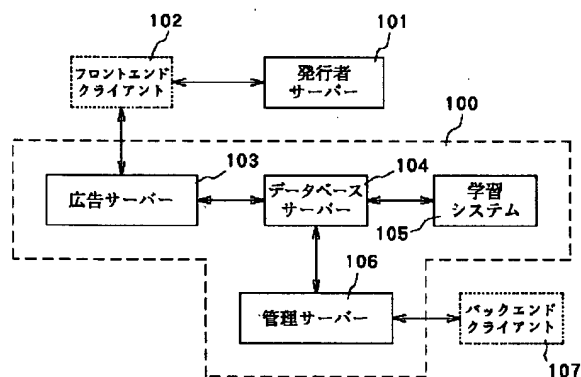
【図10】ユーザーと、動的に選択された広告を含む発行者のコンテンツサーバーとの通信例である。

【図11】カスタマイズされた広告を有する発行者のコンテンツサーバーからの結果の例である。

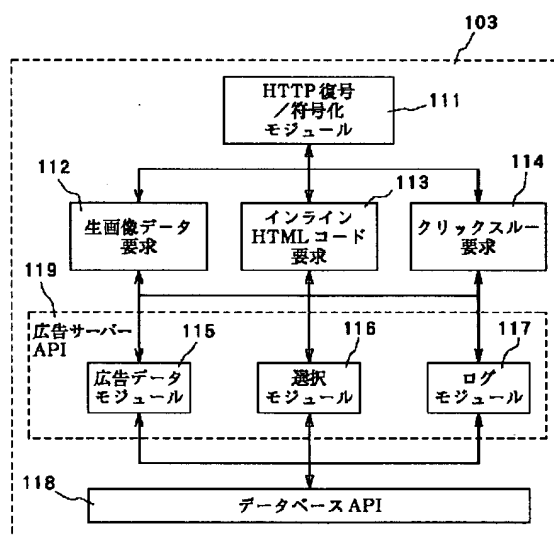
【符号の説明】

- | | |
|---------|-----------------|
| 100 | 広告システム |
| 101 | 発行者サーバー |
| 102 | フロントエンドクライアント |
| 103 | 広告サーバー (アドサーバー) |
| 104 | データベースサーバー |
| 105 | 学習システム |
| 106 | 管理サーバー |
| 107 | バックエンドクライアント |
| 111~117 | モジュール |

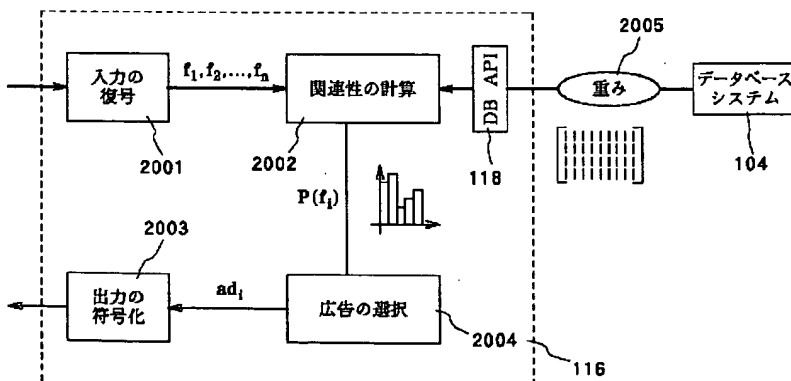
【図1】



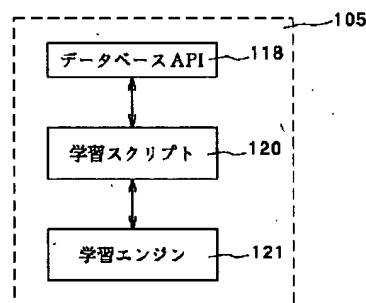
【図2】



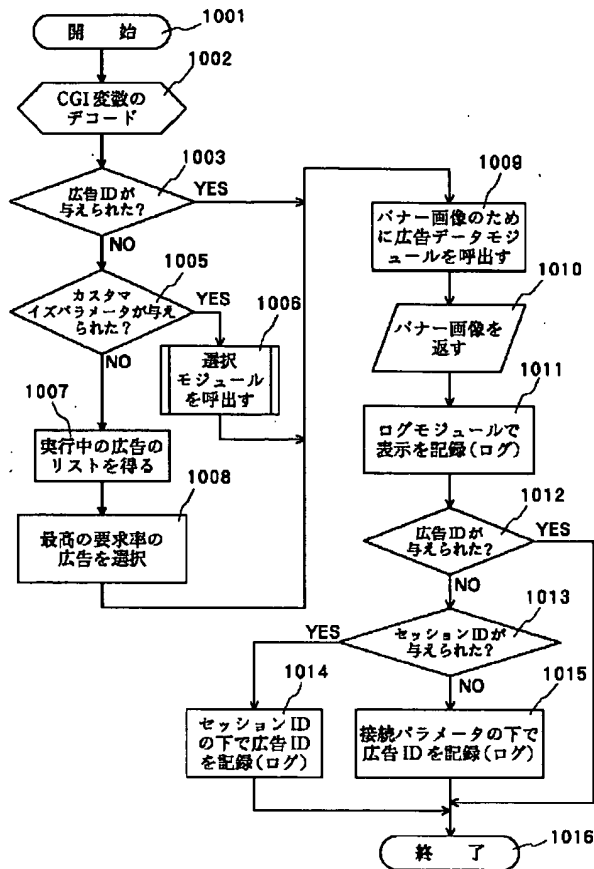
【図6】



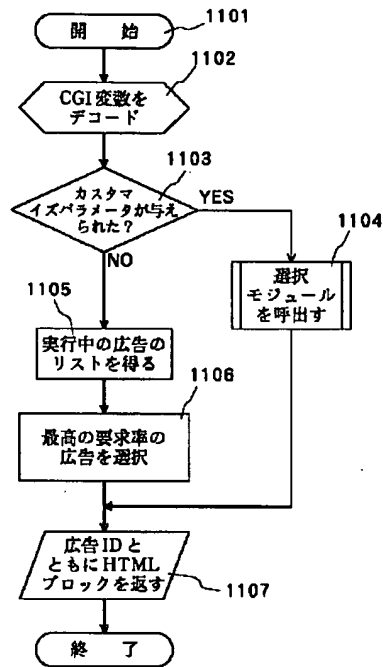
【図7】



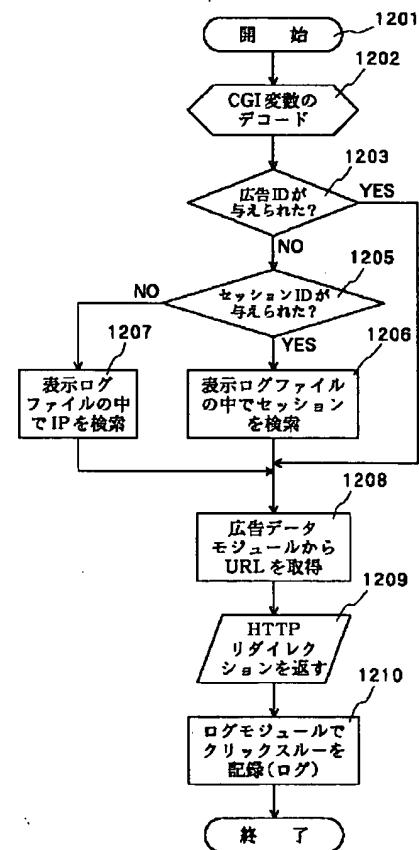
【図3】



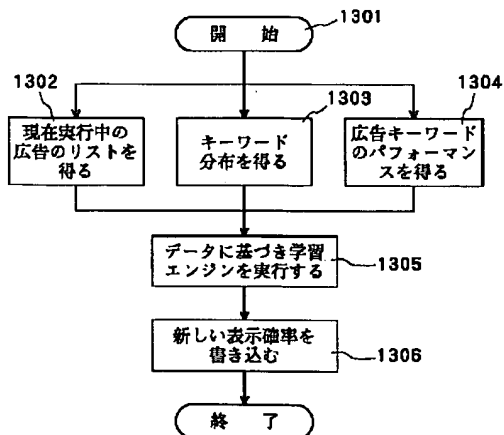
【図4】



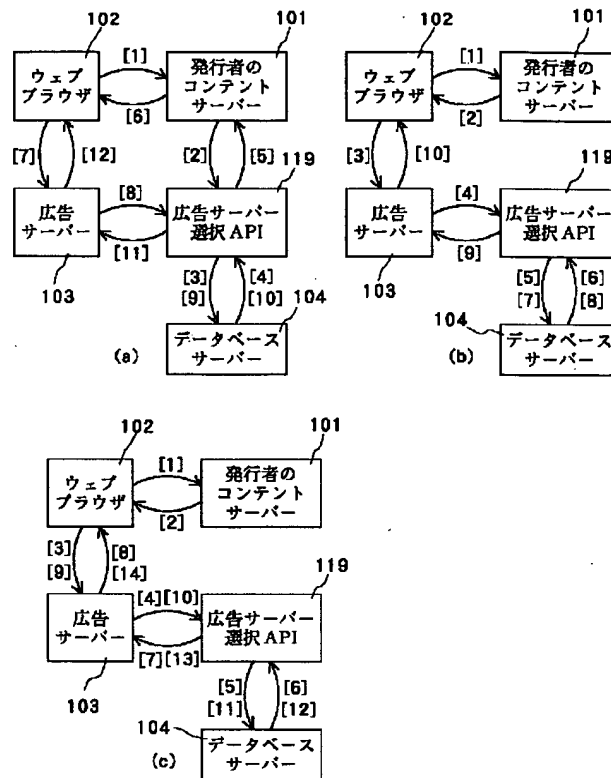
【図5】



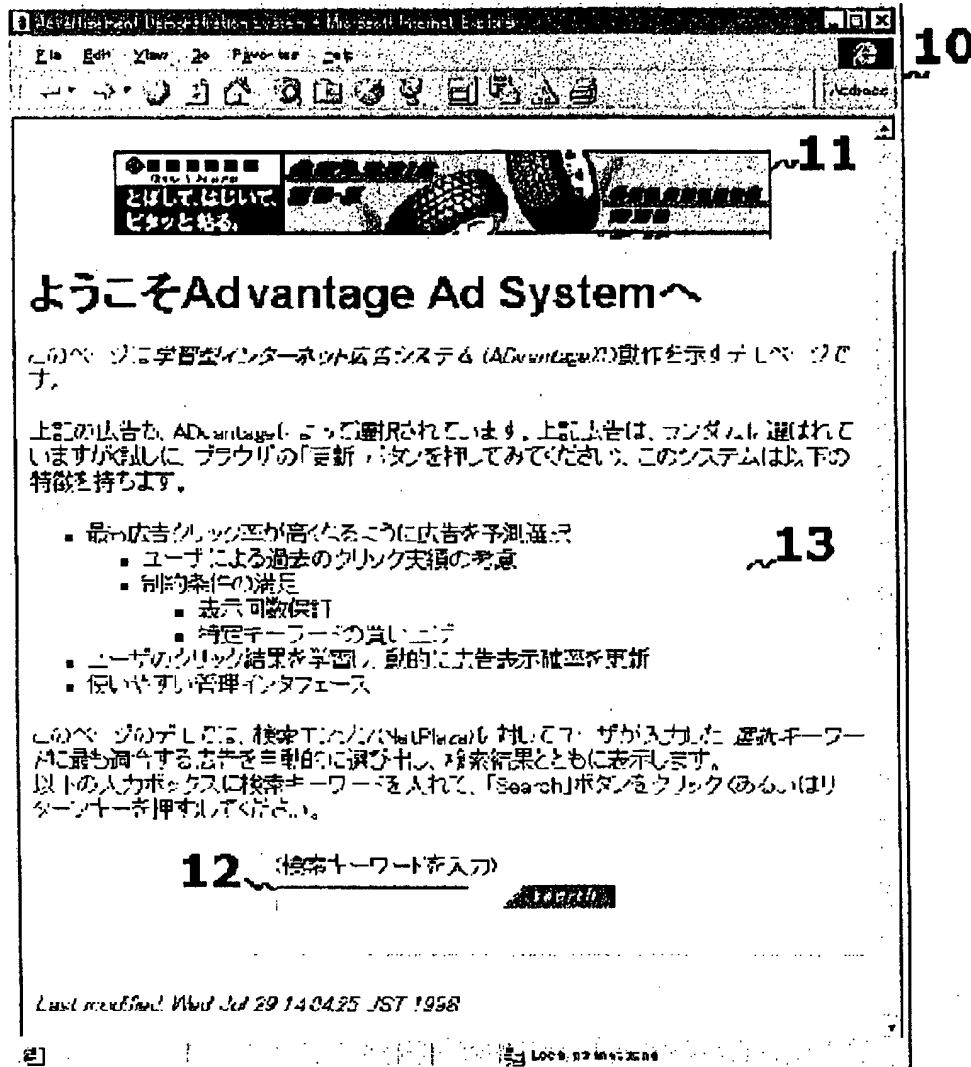
【図8】



【図9】



【 図 10 】





BEST AVAILABLE COPY